UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/819,351	03/28/2001	Katsumi Watanabe	14445	3348

23389 7590 04/22/2005

SCULLY SCOTT MURPHY & PRESSER, PC
400 GARDEN CITY PLAZA
SUITE 300
GARDEN CITY, NY 11530

EXAMINER	
STEELMAN, MARY J	

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 04/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/819,351	WATANABE, KATSUMI	
	Examiner	Art Unit	
	Mary J. Steelman	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 January 2005, 18 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-6* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 March 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *21 January 2005*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

TE

DETAILED ACTION

1. This Office Action is in response to amendment and remarks received 10/18/2004. Per Applicant's request, claims 1-6 are amended. The Specification has been amended. The Abstract has been amended. Claims 1-6 are pending.

Information Disclosure Statement

2. IDS received 28 March 2001 has not been considered. Documents are in Japanese. In reference to IDS received 02 January 2004, only the Abstract has been considered. Document is in Japanese. Applicant has submitted an English statement claiming that the following documents are relevant to the claimed invention: (Heisei 11-259335, Showa 53-41, Heisei 11-24958, Heisei 4-23740). Please submit a translated version to that Examiner may consider these prior art references. Examiner does not have ANY translated information related to Heisei 11-353205, Heisei 8-161196, or Heisei2-61731, except the following sentence: "This record of results of the prior art documents search does not constitute a reason for rejection." This is inadequate for examiner to determine the relevance of the contents of the prior art documents.

IDS received 21 January 2005 has been considered.

Drawings

3. In view of the amendment to the Specification the prior objection is hereby withdrawn.

Specification

4. In view of the amendment to the Abstract, the prior objection is hereby withdrawn.

Claim Rejections - 35 USC § 112

5. In view of the amendments to claims 1-6, the prior 35 USC 112 second paragraph rejections are hereby withdrawn.

Claim Objections

6. Examiner objects to the phrasing of claims 1 and 6 as being difficult to understand. Examiner suggests the following, which more clearly describes the three instances whereby uncompressed instruction addresses are output as trace data: (1) if a match of a predetermined instruction address or a predetermined instruction code, or (2) if a branch instruction, or (3) if trace start signal is active.

Claim 1:

1. (Currently Amended) A program development support apparatus comprising:

a CPU (Central Processing Unit) for executing a target program and outputting instruction address/instruction code data;

event management means for asserting and outputting a section trace start signal upon detecting that the instruction address/instruction code data from said CPU matches at least one of a predetermined instruction address and a predetermined instruction code set as an event condition in advance;

trace data generation means for,

when an instruction code of the instruction address/instruction code data from said CPU is a branch instruction, outputting an uncompressed instruction address as trace data;

when the section trace start signal from said event management means is active, outputting the uncompressed instruction address as the trace data;

when the event management means has detected a match of said predetermined instruction address or said predetermined instruction code, which is not a branch instruction, outputting the uncompressed instruction address as the trace data;

when the instruction address of the instruction address/instruction is not ~~the~~ a branch instruction and the section trace start signal is not active, and a match of a said predetermined instruction address of said predetermined instruction code has not been determined,

~~for~~ generating a plurality of compressed instruction addresses by compressing the instruction address of the instruction address/instruction code data, ~~and then~~ combining the compressed instruction addresses, and outputting the compressed instruction addresses as the trace data; and

a trace memory for storing the trace data from said trace data generation means. [,]

~~wherein the predetermined instruction code is different from a branch instruction.~~

6. (Currently Amended) A program development support apparatus comprising:

a trace memory for compressing and storing an instruction address that traces an instruction in a program; and

event detection means for,

upon detecting at least one of a preset predetermined instruction address and a predetermined instruction code, whereby the predetermined instruction code is different from a branch instruction,

controlling to write an instruction address, at which at least one of the predetermined instruction address and the predetermined instruction code is stored in said trace memory as uncompressed data.

(Claim 6 does not specify three conditions for writing uncompressed instruction addresses to memory: (1) if a branch instruction, (2) if start_trace signal is active, (3) if match of a predetermined instruction code segment or predetermined instruction address, that is not a branch instruction code segment. Is that intended?)

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over 5,535,331 to Swoboda et al., in view of US Patent 6,732,307 B1 to Edwards.

Per claim 1:

-a CPU (Central Processing Unit) for executing a target program and outputting instruction address/instruction code data;

(Swoboda: Col. 6, lines 41-43, "Simulator executes a software program that simulates operation of the target chip for cost-effective software development and program verification..." See fig. 44.)

-event management means for asserting and outputting a section trace start signal upon detecting that the instruction address/instruction code data from said CPU matches at least one of a predetermined instruction address and a predetermined instruction code set as an event condition in advance;

(Swoboda: Col. 6, lines 57-64, "Simulator execution modes include...While Condition Exists...Trace expressions are readily defined. In trace execution, display choices include 1)

designated expression values...Breakpoint conditions include..." Events are set as conditions or breakpoints.)

-trace data generation means for, when an instruction code of the instruction address/instruction code data from said CPU is a branch instruction, outputting... trace data; when the section trace start signal from said event management means is active, outputting trace data; when the instruction address of the instruction address/instruction code data is not the branch instruction and the section trace start signal is not active, outputting...the trace data;
(Swoboda, Abstract, lines 1-3, "Operations of a data processing device are traced (trace data generation means) by detecting a jump address (detecting branch address an event) in the program counter sequence, and pushing the jump address onto a trace stack. (assert start signal)", col. 7, lines 18-21, "A trace memory is also displayable. A record of the simulation session can be maintained in a journal file so that it can be re-executed...")

-a trace memory for storing the trace data from said trace data generation means.
(Swoboda, col. 8, lines 35-42, "Emulator controller card provides full-speed execution and monitoring of each target chip...breakpoints, software and hardware trace and timing, and single step execution are provided...Program data and program memory can be uploaded (store trace memory) or downloaded.)

-wherein the predetermined instruction code is different from a branch instruction.

(Swoboda: Col. 6, lines 57-64, "Simulator execution modes include...While Condition Exists (while predetermined instruction code is different from a branch instruction)...Trace expressions are readily defined (while predetermined instruction code is different from a branch instruction). In trace execution, display choices include 1) designated expression values...Breakpoint conditions include..." Events are set as conditions or breakpoints. Swoboda broadly disclosed that conditions/ events may be defined.)

Swoboda failed to disclose:

-outputting an uncompressed instruction address as trace data...and outputting the compressed instruction addresses as the trace data;

Swoboda failed to disclose details related to the choice of compressing or not compressing certain data prior to storing according to conditions. However, Edwards disclosed (col. 2, lines 21-21-27), "Some techniques may be used to maximize the amount of trace data gathered non-intrusively: Compress information prior to storing...instruction address,...may be compressed to save storage space..." Edwards provided the suggestion that some data may be selectively compressed after traced. Col. 7, lines 17-19, "trace may be filtered, generated selectively, and compressed to reduce the amount of space..." Edwards suggested that certain conditions (such as a start signal, a match of predetermined address / predetermined code, a branch instruction) may determine how a trace is done. Col. 10, lines 55-67, "debug circuit may provide a number of signals...may provide a number of bit values which operate as preconditions to triggering particular events in the processor. These events may then generate trace information...

watchpoint channels may cause the processor to …generate an exception. The watchpoint channels themselves may also have preconditions which determine whether or not they will…match conditions which indicate whether or not a match will occur…"

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Swoboda's processor tracing device, to include a technique to compress certain parts of the code, as disclosed by Edwards because both inventions recognized that emulation / tracing (Swoboda: col. 2, lines 15-18), "should ensure that an IC functions correctly…" thereby providing (Swoboda: col. 3, lines 21-22), "control to reduce application development time…"

Per claim 2:

-said event management means keeps a data latch signal active during a predetermined period and outputting the data latch signal;

(Swoboda: Col. 10, lines 42-43, "…bits that define control operations (like commands or instructions) available through controller card." Col. 10, lines 50-53, Shift register latches (SRLs) designated "S" are distributed through the device like a string of beads on a serial scan path…to provide access to all important registers." Col. 10, lines 65-67, "Emulation adapter in different forms involves hardwired state machine circuitry (data latch signals)…or microcoded state machine embodiments.")

-said trace data generation means receives the instruction address,/instruction code data from said CPU and the section trace start signal from said event management means and, when the data latch signal from said event management means is active, latches the instruction address/instruction code data.

(Swoboda: Col. 9, lines 1-14, "Software breakpoints allow program execution to be halted at a specified instruction address (trace start signal)…When a given breakpoint is reached, the program either halts execution to permit user observation of memory and status registers, or the breakpoint is included in a more complex condition (receive address/code data & trace start signal & data latch signal is active…), which when satisfied results in an appropriate stop mode being executed (latch instruction address/ instruction code data)…This information is suitably saved on command in a file for future analysis…")

Per claim 3:

-event setting means in which event setting data containing the predetermined instruction address/instruction code as the event condition and the active period of the data latch signal are set in advance;

(Swoboda, col. 10, lines 42-43, "…bits that define control operations available through controller card…" and lines 50-53, "Shift register latches (SRLs) designated "S" are distributed through the device…to provide access to all important registers…")

-event detection means for, upon detecting that the instruction address/instruction code contained in the event setting data output from said event setting means matches the instruction

address/instruction code of the instruction address/instruction code data from said CPU, asserting and outputting the section trace start signal and asserting the data latch signal during the active period set in said event setting means and outputting the data latch signal.

(Swoboda: Abstract, lines 1-3, "Operations of a data processing device are traced by detecting a jump address in the program counter sequence, and pushing the jump address on to a trace stack." Col. 11, lines 26-41, "The instruction register can receive serial scan signals...MUX...can select the output signal from the instruction register or from another MUX..." Col. 32, lines 51-55, "...the preferred embodiment sets up a condition in the analysis domain, and then the analysis domain effectively monitors the chip as it runs in real time, then detects when the condition occurs...")

Per claim 4:

-instruction address/instruction code latch means for latching the instruction address/instruction code data from said CPU during the active period of the data latch signal and outputting the instruction address/instruction code;

(Swoboda, col. 11, lines 26-41, "Test access port (TAP) controller is in turn coupled to instruction register (IR) and a first multiplexer. The instruction register can receive serial scan signals from the TDI line and output serially to MUX. MUX is under control of the TAP and can select the output signal from the instruction register or from another MUX. The instruction register also controls a bypass register (BR) and one or more boundary scan registers (BSR). The bypass register receives the TDA signal and outputs it to MUX. MUX is under control of the instruction register. Based on the instruction loaded into the instruction register, MUX

outputs its input from the bypass register or its input from one or more BSRs..." Col. 14, lines 2-6, "Analysis circuitry is connected to the CPU core as described more fully herein. The analysis circuitry includes condition sensors such as hardware breakpoint sensors for the controlled stops and trace stack circuitry for real-time trace recordkeeping..." Col. 31, lines 45-52, "Memory operations are suitably generated using CPU resources. Memory accesses are generated by scanning in a CPU state including appropriate CPU memory access instructions, which causes memory accesses to be generated to the appropriate memory or I/O space. This is accomplished by loading a machine state with the pipe flush bit set, and appropriate instruction in the pipeline to cause the desired memory operation.")

-branch instruction determination means for determining whether the instruction code from said instruction address/instruction code latch means is the branch instruction, and upon determining that the instruction code is the branch instruction, asserting and outputting a branch instruction detection signal;

(Swoboda: col. 3, lines 29-31, "...including a semiconductor chip, an electronic processor on-chip and an on-chip condition sensor (determine if branch instruction) connected to the electronic processor for analysis (output signal) of the operations." Col. 6, lines 51-53, "Simulation parameters are quickly stored/retrieved from files to facilitate preparation for individual sessions..." Col. 6, lines 57-64, "Simulator execution modes include...While Condition Exists...Trace expressions are readily defined. In trace execution, display choices

include...expression values...Breakpoint conditions include..." Conditions can be set to determine a branch instruction.)

-trace control means for, when the received uncompressed data selection signal is active, outputting the compressed instruction address from said instruction address data compression means as trace data, and when the uncompressed data selection signal is not active, combining a plurality of continuously received compressed instruction addresses in accordance with a bit width of said trace memory and outputting the combined instruction addresses as the trace data, and outputting a trace data write signal for instructing said trace memory to write the trace data and a trace memory address for designating a storage address of said trace memory;
(Swoboda: Col. 34, lines 42-46, "CPU core is further improved by providing a trace stack circuit distinct from program counter stack. Unlike stack, trace stack circuit develops a history (write the trace history to storage) of program counter discontinuities..." Col. 7, line 19, "...session can be maintained in a journal file (memory)...")

Swoboda failed to disclose:
-instruction address data compression means for, when a received uncompressed data selection signal is active, outputting the instruction address from said instruction address/instruction code latch means as the compressed instruction address, and when the uncompressed data selection signal is not active, outputting difference data obtained by subtracting an immediately preceding instruction address from a current instruction address as the compressed instruction address;

Swoboda failed to disclose details on compression and an active / inactive signal. However, Edwards disclosed that compression and active / inactive signals may be used in a trace apparatus. Edwards disclosed (col. 7, lines 17-19), "trace may be filtered, generated selectively (signal is active / not active), and compressed (define conditions for when to compress)..."

-OR means for asserting and outputting the uncompressed data selection signal when at least one of the branch instruction detection signal from said branch instruction determination means and the section trace start signal from said event management means is active.

Swoboda discloses a trace facility, but failed to disclose details on 'OR' means for asserting and outputting the uncompressed data selection signal when at least one of the branch instruction detection signal from said branch instruction determination means and the section trace start signal from said event management means is active. However, Edwards disclosed (col. 7, lines 17-19), "trace may be filtered (using 'OR' means), generated selectively (using 'OR' means, upon detection of branch instruction and start signal), and compressed (define conditions for when to compress, when not to compress...)..."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Swoboda's invention to trace and detect branch addresses by including details as provided by Edwards to selectively compress or not compress instructions upon the detection of certain events because both inventions recognized that emulation / tracing

(Swoboda: col. 2, lines 15-18), "should ensure that an IC functions correctly..." thereby providing (Swoboda: col. 3, lines 21-22), "control to reduce application development time..."

Per claim 5:

-frame address comparison means for asserting and outputting a frame match signal when the instruction address/instruction code contained in the event setting data output from said event setting means matches the trace memory address from said trace control means;

(Swoboda allows for control conditions (col. 10, lines 42-43), "...bits that define control operations (frame match, event setting) available through controller card." Col. 14, lines 2-6, "Analysis circuitry is connected to the CPU core...includes condition sensors such as hardware breakpoint sensors for controlled stops and trace stack circuitry for real-time trace recordkeeping...")

Swoboda failed to disclose:

-said OR means asserting and outputting the uncompressed data selection signal when at least one of the branch instruction detection signal from said branch instruction determination means, the section trace start signal from said event detection means, and the frame match signal from said frame address comparison means is active.

Swoboda discloses a trace facility, but failed to disclose details on compression and an active / inactive signal. However, Edwards suggested that some instructions/data code may remain uncompressed. The type of instruction is detected and a decision is made, according to

conditions / events, and signal asserted, whether to compress or not. Edwards: col. 10, lines 55-col. 11, line 3, "debug circuit may provide a number of signals to the processor for use in debugging operations...a number of bit values which operates as preconditions (signals, matches of predetermined code / addresses, certain types of instructions such a branch) to triggering particular events...These events may then generate trace information...watchpoint channels may cause the processor to generate ...an exception. The watchpoint channels themselves may also have preconditions (signals, matches of predetermined code / addresses, certain types of instructions such a branch) which determine...match conditions which indicate whether or not a match will occur (a match of predetermined code / address?) for a particular watchpoint, and action conditions which will determine if and what type of action occurs (compress or not compress prior to storage?) based on a watchpoint channel match."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Swoboda's invention to trace and detect conditions by including details as provided by Edwards to selectively compress instructions according to conditions because both inventions recognized that emulation / tracing (Swoboda: col. 2, lines 15-18), "should ensure that an IC functions correctly..." thereby providing (Swoboda: col. 3, lines 21-22), "control to reduce application development time..."

Per claim 6:

-a trace memory for...and storing an instruction address that traces an instruction in a program;

(Swoboda: Abstract, last sentence, "...pushing the jump address (instruction address) onto a trace stack (storing).")

Swoboda failed to disclose:

Swoboda discloses a trace facility, but failed to disclose details on compression. However, Edwards disclosed that some instructions/data code may be compressed. The type of instruction is detected and a decision is made, considering defined events and conditions defined, whether to compress or not. Edwards: col. 10, line 55-col. 11, line 3, ""debug circuit may provide a number of signals to the processor for use in debugging operations...a number of bit values which operates as preconditions (signals, matches of predetermined code / addresses, certain types of instructions such a branch) to triggering particular events...These events may then generate trace information...watchpoint channels may cause the processor to generate ...an exception. The watchpoint channels themselves may also have preconditions (signals, matches of predetermined code / addresses, certain types of instructions such a branch) which determine...match conditions which indicate whether or not a match will occur (a match of predetermined code / address?) for a particular watchpoint, and action conditions which will determine if and what type of action occurs (compress or not compress prior to storage?) based on a watchpoint channel match."

-event detection means for, upon detecting one of a preset predetermined instruction address and

predetermined instruction code, controlling to write an instruction address, in which one of the predetermined instruction address and predetermined instruction code is stored, in said trace memory as uncompressed data,

-wherein the predetermined instruction code is different from a branch instruction.

Swoboda discloses a trace facility, but failed to disclose details events / conditions and the decision whether to store trace data compressed or non-compressed. However, Edwards disclosed (col. 10, line 55-col. 11, line 3), ""debug circuit may provide a number of signals to the processor for use in debugging operations…a number of bit values which operates as preconditions (signals, matches of predetermined code / addresses that are not branch instructions, certain types of instructions such a branch) to triggering particular events…These events may then generate trace information (controlling to write an instruction address)…watchpoint channels may cause the processor to generate …an exception. The watchpoint channels themselves may also have preconditions (signals, matches of predetermined code / addresses that are not branch instructions, certain types of instructions such a branch) which determine…match conditions which indicate whether or not a match will occur (a match of predetermined code / address?) for a particular watchpoint, and action conditions which will determine if and what type of action occurs (predetermined instruction address and predetermined instruction code is stored, in said trace memory as uncompressed data) based on a watchpoint channel match."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Swoboda's invention to trace and detect branch addresses by including details as provided by Edwards because Swoboda disclosed that (col. 6, lines 57-67) 'while conditions' exist in trace expressions defined and a cache utilization policy stores trace data. Edwards merely expanded on the events / conditions that are used to define a trace and the possibility of selectively compressing or not compressing trace data to save resources

Response to Arguments

9. Applicant's arguments with respect to claims 1 and 6 have been considered but are moot in view of the new grounds of rejection.

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action.

Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3694. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

03/22/2005

TUAN DAM
SUPERVISORY PATENT EXAMINER